

## オブジェクト指向数値計算の自動負荷分散システム

鈴木 俊人 梶山 正行

tsuzuki@cis.ibaraki.ac.jp masayuki@cis.ibaraki.ac.jp

茨城大学工学部情報工学科  
茨城県日立市成沢町 4-12-1

流れの数値シミュレーションシステムはほとんどの場合、高い CPU 負荷と長大な計算時間を必要とする。そしてオブジェクト指向の数値シミュレーションシステムは、対象世界の流れ場をオブジェクトで構成しているために離散化計算単位をオブジェクトとすることができる。この点に着目し、オブジェクト指向シミュレーションなら比較的容易に分散計算にできるのではないかと考え本研究ではオブジェクト指向数値計算を PVM を使った自動負荷分散システムに発展的改造を行った。その結果、比較的容易な作業で以前の単一マシンで行った計算と同一の計算結果が得られ、計算速度もある程度の短縮が認められた。今後は更なる負荷分散システム性能の向上を目標としている。

## Automatically Distributed Load Balancing System for Object-oriented Numerical Computings

Toshihito Suzuki Masayuki Hatakeyama

Department of Computer and Information Sciences, Ibaraki University  
Nakanarusawa-cho, Hitachi-city, 316 Japan

The numerical simulations, for example the flow simulations, requires a great deals of CPU power and long computation times. To resolve the problem, some kinds of distributed numerical computing systems are needed. Then, we have developed an automatically distributed load balancing system based on the Object-oriented numerical simulation system. This Object-oriented distributed computing system has successfully been completed. The distributed computing results are good, but the computing efficiency is not always high. We will try to realize the improvements of the computing efficiencies.

### 1 はじめに

我々の研究グループではオブジェクト指向モデリングに基づく流れの数値シミュレーションシステムの研究を行なっている。それらの研究では、モデリング段階から実装段階まで一貫してオブジェクト指向モデリングの概念を用いることにより、シミュレーションの対象世界である流れ場の計算単位をオブジェクトで構成することを実現した。このようにオブジェクトで流れ場の計算単位を構成することにより、あるオブジェクトの変更によって流れ場の変更が容易な作業によって実現できる

等の多くのメリットがある。

しかし、元々莫大な計算量が必要となる流れの数値シミュレーションである故に多くの計算量が必要とするというニーズも大きい。そこで、オブジェクトという明確な計算単位があるこの数値シミュレーションは、比較的容易に分散環境に対応できるのではないかと考え、数値シミュレーションシステムを PVM [1] を用いた自動負荷分散システムへと構築することを試みた。

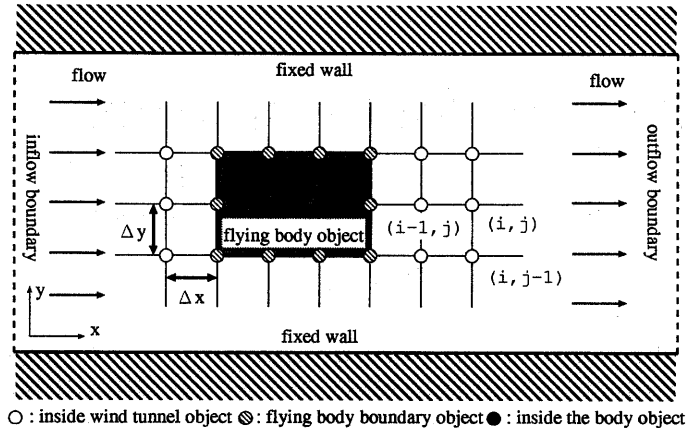


図 1: 2次元数値風洞の概要

## 2 オブジェクト指向数値計算

オブジェクト指向とは、対象世界を他と区別できるような“もの(オブジェクト)”を単位として考え、それらのモデル化された“もの”の相互作用により対象世界の振舞い(現象)を表現するモデリング方法論である。そのモデル化された“もの”のデータ(属性)とメソッド(振舞い)を強固なセットで記述することによって、従来のデータとメソッドを別々に扱ってきたものより対象世界を忠実にモデリングすることができる。また、数値シミュレーションとは数値モデルを用いて数値的に対象世界をシミュレートするものである。

従って、高いモデル化精度と柔軟なシステム変更を必要としている数値シミュレーションには、オブジェクト指向が良く合うように考えられる。特に、オブジェクトの集まりとして対象世界をモデリングしているので、対象世界の変化に対してはオブジェクトを変更すれば良いので対象世界の変化には容易に対応することが可能である。オブジェクト指向数値シミュレーションにとって最も大きな障害となるのは、計算量の増加である。元々、数値シミュレーションは、莫大な計算量を必要としている。

実際のオブジェクト指向モデリングに基づく数値シミュレーションを説明するために以前構築した2次元の数値風洞 [2] を例にあげる(図 1参照)。対象世界である風洞を $\Delta x, \Delta y$ ごとに直交格子によって分割し、その交点である計算点をオブジェクト

とする。各オブジェクトは、データとメソッドの1セットを持っており、それぞれ以下の通りである。

- データ
  - 計算点(メッシュの交点)での物理量(密度、速度、圧力等)
  - 上下左右のオブジェクトとの距離
  - 上下左右のオブジェクトとのリンク
  - 自身のオブジェクト種類(物体内オブジェクト、物体外オブジェクト等)
  - etc.
- メソッド
  - 差分化した流れの支配方程式(Navier-Stokes 方程式等)
  - データを返す関数(メッセージ通信)
  - etc.

計算は、各オブジェクトがある時間ステップでの自分や周りのオブジェクトの物理量を参照して次の時間ステップの物理量を計算し、すべてのオブジェクトの物理量が求まったら、一斉に物理量を更新し再び計算をはじめ。上記の計算過程を繰り返し、時間ステップを進めていくことでシミュレーションが進んでいく。

## 3 自動負荷分散システム

自動負荷分散システムを構築するために、分散通信ライブラリであるPVM(Parallel Virtual Machine) [1] を使ったマスター/スレーブ方式を採用して実現する。

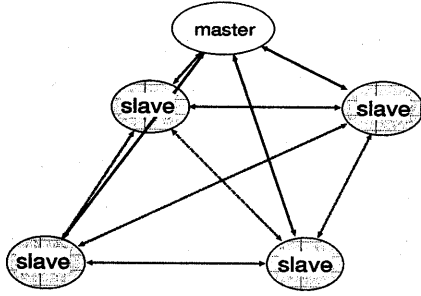


図 2: マスター/スレーブ方式の分散システム

マスター/スレーブ方式は、図 2 のような構成で実際の計算を行うスレーブと各スレーブを管理するマスターからなる。マスター/スレーブ方式では、管理側(マスター)と計算側(スレーブ)にプログラムが 2 分化されているので SPMD (Single Program Multiple Data) 方式等に比べて設計が簡単になる。スレーブの起動については、マスターが起動し、その後のスレーブの管理、消去等もマスターが一貫して行う。

本研究で用いたマスター/スレーブ方式を使った自動負荷分散システムでは、スレーブから計算結果と共に送られてきた計算所要時間を元に、マスターが次の時間ステップでスレーブに割り当てる計算の量を変更し、各スレーブにかかる時間の差を小さくしていく(図 3 参照)。

#### 4 オブジェクト指向数値計算と分散計算

オブジェクト指向数値計算(シミュレーション)では、対象世界をすべてオブジェクトとしている。そのため、計算単位がオブジェクトとなっている。計算点でもあるオブジェクトは、その点における物理量等をデータとして、そして自分自身の次の時間ステップにおける物理量を求める計算式をメソッドとして持っている。そのため、各オブジェクトは独立に計算を行うことができ、自分の物理量を更新することもできる。従って、オブジェクトのように独立していて、自分のデータを書き換えることができるメソッドを持ち、かつ自分のデータは自分でしか書き換えられない(カプセル化)計算単位は複数台のマシンに任意の数のオブジェク

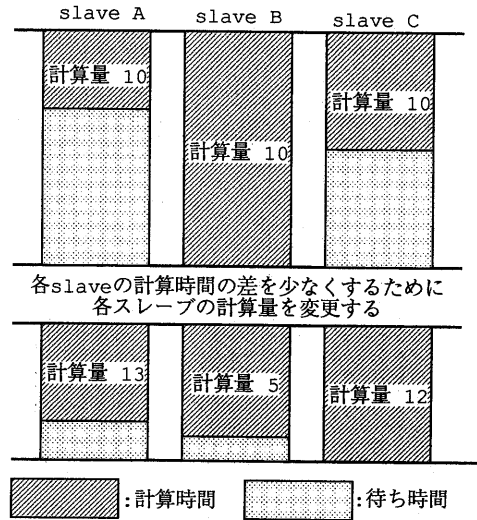


図 3: 自動負荷分散システム

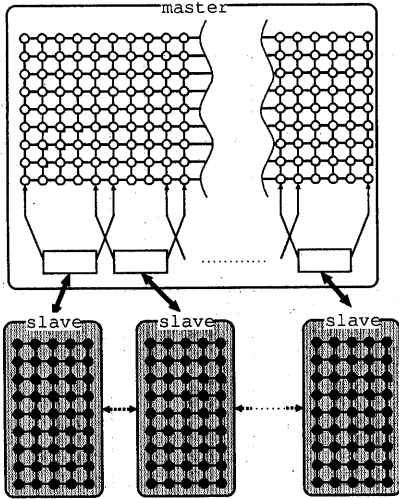
トを分散させて扱えば従来の分散処理システムよりも簡単に実現できるのではないかと考えられる。

そこで、本研究では複数のオブジェクトをまとめてひとつのオブジェクトとして扱い、そのオブジェクトの集合体(これを 1 つのオブジェクトと考えて)を 1 つのスレーブに割り当てる形にした。実際には全オブジェクトをマシンの台数分に分割して、各オブジェクトの集合体を構成した。

#### 5 計算の概要

本研究で構築した 2 次元の連続流体の数値シミュレーションの自動負荷分散システムの構成を図 4 に示す。マスターは、シミュレーションの対象世界全体の計算点について把握している必要があるので各計算点に対応した Node オブジェクトを、またマスターは各スレーブも管理する必要があるため各スレーブに対応した Cell オブジェクトを持っている。そして、スレーブはマスターによって決められたシミュレーションの対象世界の一部分の情報を保持している必要があるため、各計算点に対応した Grid オブジェクトを持っている。

以下にそれぞれのクラスオブジェクトについて説明する。



○:Node object □:Cell object ●:Grid object

図 4: 2次元の連続流体の数値シミュレーションの自動負荷分散システムの構成

### 5.1 Node オブジェクト

Node オブジェクトは、マスターがシミュレーションの対象世界全体の計算点についての情報を保持するためのオブジェクトである。データとしては、その計算点における物理量や上下左右の計算点へのリンク等を持っている。メソッドとしては、物理量等のデータを送受信する関数等を持っている。スレーブの方で数値計算は行うのでこのオブジェクトのメソッドとして計算に関するものはない。図 5 に Node オブジェクトの OMT 図 [3] を示す。

### 5.2 Cell オブジェクト

Cell オブジェクトは、マスターが各スレーブを管理するため、また各スレーブに関する情報を保持するためにあるオブジェクトである。実世界には、存在しないオブジェクトでシミュレーションを行う際必要となった仮想的なオブジェクトである。データとしては、以下のようなものがある。

- スレーブが担当する計算範囲の最初と最後へのリンク (つまり、Node オブジェクトのリンク)

- マスターがスレーブを起動したとき、分散環境内で一意に決定されるプロセスの id
- スレーブが動いているマシンのホスト名
- スレーブが一定回数の計算が終るまでにかかった時間
- スレーブが担当している計算領域の計算点数
- etc...

メソッドとしては、上記のデータを送受信する関数等を持っている。図 6 に Cell オブジェクトの OMT 図を示す。

### 5.3 Grid オブジェクト

Grid オブジェクトは、計算を行うためのオブジェクトである。分散環境内で実際に計算を行うのが、このオブジェクトである。データとしては、マスターの Node オブジェクトと同様のその計算点における物理量や上下左右の計算点へのリンク等を持っている。メソッドも Node オブジェクトとほとんど同じだが、計算を行うために次の時間ステップにおける自分の物理量を求める式 (差分化した Navier-Stokes 方程式) を持っている。図 7 に Grid オブジェクトの OMT 図を示す。

### 5.4 計算の流れ

計算は、以下の手順で行う。

1. マスターにある計算点 (Node オブジェクト) の初期設定。物理量の設定等。
2. マスターがスレーブの担当する計算範囲の計算点 (Node オブジェクト) の状態 (物理量など) や隣接するスレーブの情報 (プロセス id 等) を渡し、それを受けとったスレーブが自分の担当範囲の計算点 (Grid オブジェクト) の状態などを設定する。
3. 次ステップの物理量を求める計算を行なう。
4. スレーブ同士で境界の物理量のやりとりを行なう。
5. 3.~4. を一定回数繰り返す。
6. 各スレーブが、マスターに計算結果を渡す。マスターは、受けとった計算結果を出力する。そして、スレーブごとの計算時間から計算点の割り当て数の変更を行なう。
7. 3.~6. をある回数繰り返す。
8. ある回数繰り返し終ったら、計算結果を出力し、計算終了。

このような流れで計算を行っていく。

Node
<ul style="list-style-type: none"> <li>・物理量</li> <li>・隣接する計算点ヘリンク</li> <li>・etc</li> </ul>
<ul style="list-style-type: none"> <li>・物理量を送信する関数</li> <li>・物理量を受信する関数</li> <li>・etc</li> </ul>

図 5: Node オブジェクトの OMT 図

Cell
<ul style="list-style-type: none"> <li>・担当する計算領域の最初と最後の計算点ヘリンク</li> <li>・プロセスのid</li> <li>・マシンのホスト名</li> <li>・数ステップ間の計算時間</li> <li>・担当する計算領域の計算点数</li> <li>・etc</li> </ul>
<ul style="list-style-type: none"> <li>・スレーブに関する情報を設定する関数</li> <li>・スレーブに関する情報を返す関数</li> <li>・etc</li> </ul>

図 6: Cell オブジェクトの OMT 図

Grid
<ul style="list-style-type: none"> <li>・物理量</li> <li>・隣接する計算点ヘリンク</li> <li>・隣接する計算点までの距離</li> <li>・etc</li> </ul>
<ul style="list-style-type: none"> <li>・各データを送信する関数</li> <li>・各データを受信する関数</li> <li>・次ステップの物理量を計算する関数</li> <li>・etc</li> </ul>

図 7: Grid オブジェクトの OMT 図

## 5.5 スレーブへの計算範囲割り当て方法

スレーブに割り当てる分散オブジェクトの単位は、 $y$  方向の 1 列ごとの計算点の集まりとする。これは、スレーブ間で境界の情報をやりとりするときの通信量を減らすためである。最初にマスターは、各スレーブに均等に計算範囲を割り当てる。その後の計算範囲の変更は、スレーブの計算時間とスレーブに割り当てられた計算範囲内にある計算点数に関係してくる。

まず、計算範囲の変更を行う必要があるかどうかの判断は、単純にスレーブから返ってくる計算時間の比較によって行っている。全スレーブの計算時間の最大値が最小値の一定倍数以上のときのみ計算範囲の変更を行うようにしてある。次に割り当て方法だが、スレーブの計算時間とスレーブに割り当てられている計算点数から 1 つの計算点あたりの計算時間を求め、その 1 つの計算点あたりの計算時間が短いスレーブには前よりも広い計算範囲になる様に、長いスレーブには狭い計算範囲になる様に計算範囲を変更する。そうすることによって、スレーブごとの計算終了時間の差を少なくしようと試みる。

## 6 計算結果

以下のような条件で計算を行う。

- ・格子は等間隔で、幅  $\Delta x = 0.01$ 、 $\Delta y = 0.01$  で  $0.00 \leq x \leq 1.00$ 、 $0.00 \leq y \leq 0.50$ 。時間幅

は  $\Delta t = 0.0005$  で 1000 ステップ計算した。

- ・初期状態は、密度  $\rho = 1.0$ 、速度  $u = 0.0$ 、 $v = 0.0$ 、エネルギー  $e = 1.8$ 、圧力  $p = 1.0$ 、レイノルズ数  $Re = 500$ 、マッハ数  $M = 0.4$ 。
- ・境界条件は、 $x = 0.00$  では一定で、 $x = 1.00$  では  $x = 0.99$  と等しい物理量にしている。また、 $y = 0.00$  では  $y = 0.01$ 、 $y = 0.50$  では  $y = 0.49$  と等しい物理量にしている。物体境界は、 $u$ 、 $v$  共に  $0.0$  として、その他の物理量はひとつ流体側と等しくした。
- ・物体は、矩形で  $0.60 \leq x \leq 1.00$ 、 $0.00 \leq y \leq 0.08$  の範囲に挿入してある。
- ・分散計算を行なう場合、2 台のマシンに 1 つずつスレーブを立ちあげ、100 ステップごとに計算時間による計算範囲の変更を行なうかどうかを決める。

このような条件で行なった、単一マシンでの計算結果と分散環境での計算結果を図 8 と図 9 に示す。

また表 1 は、それぞれの計算にかかった計算時間である。この時間は、10 回計算を行ない、その平均値、最大値、最小値を求めたものである。



図 8: 単一マシンで行なった計算結果の密度分布

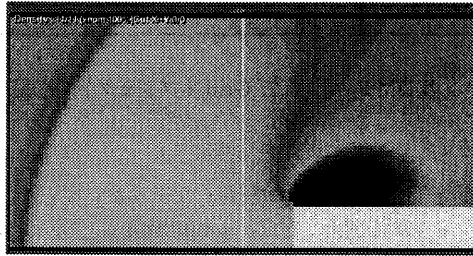


図 9: 2 台のマシンで行なった計算結果の密度分布

表 1: 計算時間の比較 (単位:秒)

	単一	分散
real time ( average )	356	259
real time ( maximum )	410	303
real time ( minimum )	335	246

## 7 考察

計算結果から見て分かる通り、単一マシンで計算を行っても、分散環境で複数のマシンで行っても計算の結果は変わらないことから計算手法の正しさは立証されている。ただし、物体の流れの影響は上流方向には風洞最上流まで及んでいない。衝撃波の様にも見える波が波及の最上流である。分散環境で計算を行う場合、スレーブの境界(マシンの境界)で値の変化等を防ぐためにスレーブ間の同期をうまくとることで回避できた。しかし、そのため表 1 を見ても分かる通り計算時間の短縮については分散計算をした効果があまり大きくは現れていない。今後は、分散計算での計算時間の短縮を目指していきたいので、自動負荷分散の仕方に工夫を加えていきたい。

本研究では、以前に構築したオブジェクト指向数値シミュレーションシステムを有効に活用して実装した。しかしこれが、手続き型言語で組まれたシステムであったら本研究のように容易に構築しにくい可能性があることを指摘したい。それは、分散システムに発展させるためにはアルゴリズム的に並列的な処理が可能な部分が容易に再構成できなければならない。しかし、手続き型言語で組まれたシステムは、はじめから並列に処理を行うという考えがなければ、並列的な処理が可能な部分を抽出することは煩雑あるいは困難であることはよく知られている。これに対しオブジェクト指向数値シミュレーションシステムでは、オブジェクトがすでにそれ自身で独立(並列)に計算を行えるのでそのまま分散システム用の単位として活用できる。このような理由で、本研究では比較的簡単にオブジェクト指向シミュレーションを利用して自動負荷分散システムが構築できたことが分かった。

## 8 まとめ

本研究では、オブジェクト指向数値計算のPVMを使った自動負荷分散システムを構築して評価した。しかし、まだ分散環境におけるメリット(計算時間の短縮等)があまり明確な結果として出てきていないので、今後に自動分散の仕方等に工夫が望まれる。

## 参考文献

- [1] Al Geist 他 著、村田英明 訳、“PVM3 ユーザーズガイド&リファレンスマニュアル日本語版” 1994年8月。
- [2] 畠山正行、鈴木俊人、“オブジェクト指向に基づく柔軟な数値風洞シミュレーションシステム” 情報処理学会第52回全国大会講演論文集、pp.5-71~5-72、1996年3月6日。
- [3] J.ランボー 他 著、羽生田栄一 訳、“オブジェクト指向方法論 OMT” トッパン、1992年。